Theoretical Computer Science 112 (1993) 311–337
Elsevier

Deciding whether a finite set of words has rank at most two

Jean Néraud

LIR, LITP, Institut Blaise Pascal, Université de Rouen, Faculté des Sciences, Place Emile Blondel, F-76134 Mont-Saint-Aignan, France

Communicated by D. Perrin
Received January 1991
Revised June 1991

Abstract

Néraud, J., Deciding whether a finite set of words has rank at most two, Theoretical Computer Science 112 (1993) 311–337.

Given a finite subset X of a free monoid A^*, we define the rank of X as $r(X)=\min\{|Y|: X\subseteq Y^*\}$. The problem we study here is to decide whether or not $r(X)\leqslant 2$. We propose an $O(n\ln^2 m)$ algorithm, where n stands for the sum of the lengths of the words in X, and m stands for the length of the longest word.

0. Introduction

In the context of the free-monoids theory, two notions of rank are actually proposed. First, given a set of words X of the free monoid A^*, its rank can be defined as the cardinality $r_1(X)$ of the basis of the free hull of X, i.e. the smallest free submonoid which contains X. In another way, we can define the rank of X as the smallest cardinality $r(X)$ of a finite set Y satisfying $X\subseteq Y^*$, i.e. such that all the words in X can be written as the concatenation product of words in Y.

The first notion of rank was illustrated by the famous defect theorem (cf. e.g. [9]), which says that given a finite $X\subseteq A^*$, if X is not an unique decipherable code (or, for short, a code) then its cardinality $|X|$ satisfies $r_1(X)\leqslant|X|-1$. In other words, X is a code iff $r_1(X)=|X|$.

The second notion corresponds to the concept of degree, introduced in [7]. This topic meets that of elementariness: a finite subset X is elementary (or independent) iff

Correspondence to: J. Néraud, LIR, LITP, Institut Blaise Pascal, Université de Rouen, Faculté des Sciences, Place Emile Blondel, F-76134 Mont-Saint-Aignan, France. Email: neraud@litp.ibp.fr.

$r(X)=|X|$. Historically, the notion of elementariness applied to morphisms, and its introduction constituted the major step in the delicate proof of the decidability of the famous DOL sequence equivalence problem.

Although these two notions of rank seem very close, their algebraical and topological properties are different, as shown in [7, 11, 12]. Another important difference concerns the decision problems. It is well known that, given a finite set of words X, deciding whether X is a code can be achieved by applying the classical Sardinas and Patterson algorithm (cf. [2, p. 50]). This algorithm has been implemented so that it allows to process the set X in time $O(n \ln |X|)$ (cf. [14, 1]), where n stands for the sum of the lengths of the words in X, and $|X|$ stands for the cardinality of X. Moreover, the computation of $r_1(X)$ may be done in time $O(|X|n^2)$, as shown in [15, 16].

On the contrary, deciding whether X is elementary, or deciding whether $r(X)$ is smaller than a given integer k, are NP-hard problems [10]. In this way, it is of interest to examine restrictions of this last problem. For instance, consider a set X whose elements are words with length p, where p is a given positive integer. If $p=2$ then computing $r(X)$ can be done in time $O(|X||A|)$, by determining the family of the connected components of a direct graph associated with X. However, for $p \geqslant 3$, deciding whether $r(X) \leqslant k$ remains NP-complete. As another example, given a finite set of words X, deciding whether $r(X)=1$ can be done by applying the algorithm of Knuth et al. [8] (indeed, we have $r(X)=1$ iff $r_1(X)=1$). The corresponding complexity is a linear function of the sum of the words in X.

In this paper, we are interested in the following restriction: given a finite set of words X, the problem is to decide whether or not $r(X) \leqslant 2$. This new restriction is justified by the fact that, from a theoretical point of view, many combinatorial properties has been established in the framework of two-element codes (cf. e.g. [17, 5]). The naive algorithm consists in examining all the two-element subsets of factors of the words in X. For any subset $\{x, y\}$, we shall decide whether or not $X \subseteq \{x, y\}^*$. This method leads to an $O(n^5)$ algorithm. A refinement in $O(n^3)$ can be done by considering restrictive properties of prefixity on $\{x, y\}$. Here, we establish the following theorem.

Theorem 0.1. *Given a finite set $X \subseteq A^*$, deciding whether $r(X) \leqslant 2$ or not can be achieved in time $O(n \ln^2 m)$, where n stands for the sum of the lengths of the words in X, and m stands for the length of the longest word.*

Our proof makes use of a result of [11], which allows to restrict to biprefix sets $Y=\{x, y\}$ with primitive elements (i.e. whose elements are only trivial powers of other words). Actually, given a finite set X, we shall conjugate the preceding property with results on the periods [6] and repetitions [4] in a word. We shall construct a set with cardinality $O(\ln^2 m)$, namely $TEST(X)$, and which satisfies the following property:

$r(X)=2$ iff there exists a pair of words $(\alpha, \beta) \in TEST(X)$ with $X \subseteq \{\alpha, \beta\}^*$.

We now shortly describe the contents of our paper.

Section 1 is concerned with the basic definitions. The terminology of free monoids is settled, and we recall some properties of the so-called overlaps of a word.

In Section 2, we introduce the general problem of the computation of the rank, and we consider the restriction which concerns our paper. We also present the main feature of our algorithm. Given a finite set X, it consists in computing the preceding set $TEST(X)$.

The rest of our paper explains the different steps of the construction, whose main feature is the following:

- In a first step, we construct a two element set of factors of X, namely $ABSTRACT(X)$, such that for every two element biprefix set Y, if $X \subseteq Y^*$ then necessarily we have $ABSTRACT(X) \subseteq Y^*$. This allows to restrict our search of the "candidates sets" Y (cf. Section 3).
- In a second step, given a two-element code Y, we examine the possible cases of factorization of $ABSTRACT(X)$ over Y. This leads to collect necessary conditions, which apply to pairs of words (x, y) satisfying $X \subseteq Y^*$, with $Y = \{x, y\}$ (Sections 4 and 5).
- In Section 6, we establish successive refinements of the preceding conditions. These refinements lead to the construction of our set $TEST(X)$.

1. Preliminaries

1.1. Definitions and notation

Given a finite alphabet A, we denote by A^* the free monoid it generates and by ε the word of length 0. We set $A^+ = A^* - \{\varepsilon\}$. For any arbitrary subsets $X, Y \subseteq A^*$, we denote by XY their (concatenation) product, by X^* the submonoid generated by X (we set $X^+ = X^* - \{\varepsilon\}$), and by XY^{-1} the set $\{u \in A^* : \exists (x, y) \in X \times Y \; x = uy\}$. The *rank* of X is the integer: $\min\{|Y| : X \subseteq Y^*\}$. Given a word $w \in A^*$, we denote by $pref(w)$ ($suff(w)$) the set of all *prefixes* (*suffixes*) of w, i.e. the set of all words u satisfying the condition: $w \in uA^*$ ($w \in A^*u$). We say that u is an *interior factor* of w iff $w \in A^+uA^+$. Given two words w, w', we denote by $w \wedge w'$ the longest common prefix of w and w'. If $w = w_1 \dots w_n$, with $w_i \in A$ $(1 \leqslant i \leqslant n)$, its *reversed* word is $\tilde{w} = w_n \dots w_1$.

For every subset $X \subseteq A^*$, we set $pref(X) = \bigcup_{w \in X} pref(w)$, $suff(X)$ being defined in a similar way. We say that X is *biprefix* iff $X \cap XA^+ = A^+X \cap X = \emptyset$. In other words, X is biprefix iff for any word $x \in X$, there exists no word $y \in X \setminus \{x\}$ with x a prefix or a suffix of y.

Given a word $w \in A^+$, w is primitive iff $w = x^n$ implies $n = 1$. Otherwise, w is called imprimitive.

1.2. Overlap of a word

Given a word $w \in A^+$, any word $x \neq w$ which is both prefix and suffix of w is called an *overlap* of w, and the integer $|w| - |x|$ is called a *period* of x. The overlaps of a word are in one-to-one correspondence with its periods [9].

Given a word, the connection between its different overlaps is described by the following classical theorem (cf. [9, p. 10]):

Theorem 1.1 (Fine and Wilf). *Let $x, y \in A^*$ such that two powers of x and y have a common prefix of length $|x|+|y|-\gcd(|x|,|y|)$. Then x and y are powers of the same word.*

We denote by φ the function which, with every word $w \neq \varepsilon$ associate the longest overlap of w. The computation of $\varphi(w)$ can be done in time linear in $|w|$ by applying the algorithm of Knuth et al. (KMP-algorithm: cf. [8]). This algorithm applies the following classical recursive rule [13]:

Rule 1. Let $p \in pref(w)$ and let $a \in A$

$$\varphi(pa) = \begin{cases} \varphi(p)a & \text{if this word is a prefix of } w, \text{ otherwise:} \\ \varphi(\varphi(p)a) & \text{if } p \neq \varepsilon \\ \varepsilon & \text{otherwise.} \end{cases}$$

Let (y, z) be the unique pair of nonempty words such that $w = y\varphi(w) = \varphi(w)z$. Clearly, the words y and z are conjugate (cf. e.g. [9, p. 8]). Moreover, there exists a unique pair of words $(u, v) \in A^+ \times A^*$ which satisfies:

(1.1) – $y \in uv$, $z = vu$, $x \in (uv)^* u$
– uv is a primitive word (indeed, $\varphi(w)$ is the longest overlap of w)

1.3. Primitive words

Given a word $w \in A^+$, w is *primitive* iff $w = x^n$ implies $n = 1$ otherwise w is *imprimitive*.

The following remark is a direct consequence of the defect theorem (cf. [9, p. 6]). It will be of common use in our paper.

(1.2) *If x is a primitive word then x cannot be an interior factor of x^2.*

As a consequence of Fine and Wilf's result, given a word w, there exists an unique word z such that $w \in z^+$: z is called the *primitive root* of w.

Given a word $w \in A^+$, the KMP-algorithm leads to an $O(|w|)$-time computation of its primitive root z (cf. Section 1.1) of w. Indeed, z is the unique word which satisfies the following condition:

(1.3) – If $\varphi(w) \neq \varepsilon$ and if $|w| - |\varphi(w)|$ divides $|w|$, then we have $z = w.(\varphi(w))^{-1}$.
– Otherwise, we have $z = w$.

2. Computation of the rank

2.1. The problem

In this paper, we shall study a restriction of the general problem *RANK*, whose terms are the following:

Instance: A finite set of words X, and an arbitrary integer $k \in \mathbb{N}$.

Question: Is $r(X)$ not greater than k?

As established in [10], this problem is NP-complete.

The case of sets of rank 1

Given a finite set of words X, it is easy to decide whether or not $r(X)=1$. Indeed, we shall apply the following algorithm.

Algorithm 2.
begin
$w \leftarrow$ a word in X;
apply KMP algorithm for computing the primitive root of w, namely z;
if all the words $w' \in X \setminus \{w\}$ belong to z^* **then** "$r(X)=1$" **else** "$r(X) \geqslant 2$"
end

Clearly, the complexity of Algorithm 2 is a linear function in the sum of the lengths of the words in X.

In our paper we shall study the following new restriction of $RANK$:
Instance: A finite set of words $X \subseteq A^+$, with $r(X)>1$.
Question: Decide whether or not $r(X)=2$.

(Clearly, we may assume that the basic alphabet, A, satisfies $|A| \geqslant 3$.)

2.2. A property of biprefixity

Let X be a finite subset of A^*. According to [3], there exists a biprefix set Y such that $X \subseteq Y^*$, and $|Y| \leqslant |X|$. Moreover, in [11], this result is extended to obtain the following:

Proposition 2.1. *Given a finite set $X \subset A^*$, there exists a biprefix set $Y \subseteq A^*$, which satisfies the three following conditions:*
(1) *All the elements of Y are primitive words;*
(2) $X \subseteq Y^*$;
(3) $r(X)=|Y|$.

2.3. The main feature of our algorithm

It is now convenient to explain the scheme of our method, which is based on the following result.

Theorem 2.2. *Given a finite set of words $X \subseteq A^+$, there exists a finite set $TEST(X) \subseteq A^+ \times A^+$, such that the following holds:*
(1) $|TEST(X)| \sim O(\ln^2 m)$, *with* $m=\max\{|w|: w \in X\}$;
(2) *for all the pairs $(\alpha, \beta) \in TEST(X)$, the set $\{\alpha, \beta\}$ is biprefix, and α, β are primitive words;*
(3) $r(X)=2$ *iff there exists a pair $(\alpha, \beta) \in TEST(X)$ such that $X \subseteq \{\alpha, \beta\}^*$.*

This result leads to a decision algorithm (Algorithm 3), which is described below.

In the corresponding implementation, words will be represented by linked lists. We shall represent pairs of words and sets of words by linked lists of words. Given two sets of words E, F, their union $E \cup F$ will be represented by the concatenation of the corresponding linked lists. Clearly, in the resulting list, multiple copies of the same element may appear. This does not affect the orders of the cardinalities, indeed we only make use of the inequality: $|E \cup F| \leqslant |E| + |F|$.

Algorithm 3.
begin
(1) compute the set $TEST(X)$ of Theorem 2.2;
(2) **for** all the pairs $(\alpha, \beta) \in TEST(X)$ **do** decide whether or not $X \subseteq \{\alpha, \beta\}^*$.
end

First, it is convenient to explain step 2.

2.4. *The phase of converse in Algorithm 3*

Given a pair $(\alpha, \beta) \in TEST(X)$, deciding whether $X \subseteq \{\alpha, \beta\}^*$ is easily done by constructing the "flower" automaton (cf. [2, p. 189]) with behavior $\{\alpha, \beta\}^*$, and by deciding whether or not this automaton may recognize all the words in X.

Since $\{\alpha, \beta\}$ is a prefix set, our automaton is deterministic and it has exactly $|\alpha| + |\beta| - |\alpha \wedge \beta|$ states (cf. Fig. 1). Moreover, from a given state, at most two arrows may start. Hence, the construction requires time and space $O(|\alpha| + |\beta|)$; thus, it is done in time $O(\max\{|x|: x \in X\})$.

Deciding whether a word $w \in A^*$ belongs to the behavior of the preceding automaton will be done in time $O(|w|)$, which is independent of the size of the alphabet, $|A|$ (indeed, each step consists in at most two comparisons).

Consequently, in step 2 of Algorithm 3, deciding whether $X \subseteq \{\alpha, \beta\}^*$ will be done in time $O(\sum_{w \in X} |w|)$.

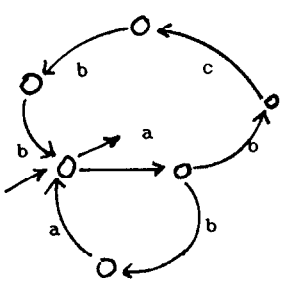

Fig. 1. Flower automaton with behavior $\{abcbb, aba\}^*$.

The rest of our study will consist in the following steps:
- proof of Theorem 2.2, and effective construction of the set $TEST(X)$.
- algorithmic interpretation of the preceding construction. We shall establish that the computation may be done in time $O((\sum_{w\in X}|w|)\ln^2 \max_{w\in X}|w|)$.

3. A first restriction of the search: set $ABSTRACT(X)$

According to Proposition 2.1, if $r(X)=2$, there exists a two-element set $\{x,y\}$, with primitive elements, such that $X\subseteq\{x,y\}^*$. The main feature of the construction of our set $TEST(X)$ is to collect necessary conditions on the pair $\{x,y\}$.

In a first step we shall construct a two-element subset of factors of X, namely $ABSTRACT(X)$, which satisfies the following condition:

For all the biprefix sets $Y\subseteq A^+$, if $X\subseteq Y^+$ then $ABSTRACT(X)\subseteq Y^*$.

3.1. Notation ABSTRACT( )

Lemma 3.1. *Given a finite set $X\subseteq A^+$, with $r(X)\geqslant 2$, there exists a two-element subset Z of X, such that $r(Z)=2$.*

Proof. Let $w_1, w_2\in X$, and let y be the primitive root of w_1. If $r(\{w_1,w_2\})=1$, then we have $w_2\in y^*$. Assume that, for all the two-element subsets $Z\subseteq X$, we have $r(Z)=1$. Clearly, for all the words $w\neq w_1$, y is the primitive root of w. Consequently, we have $X\subseteq y^*$, which contradicts $r(X)\geqslant 2$. □

Clearly, with the notation of Lemma 3.1, if $X\subseteq\{x,y\}^*$, then we have $Z\subseteq\{x,y\}^*$.

Lemma 3.2. *Given a two-element set $Z\subseteq A^+$, with $r(Z)=2$, there exists another two-element set T, which satisfies the following conditions:*

(1) *T is a biprefix set;*

(2) *for all the biprefix set Y, if $Z\subseteq Y^*$ then $T\subseteq Y^*$.*

Proof. Given a two-element set $U\subseteq A^+$, we set

$$\sigma(U)=\begin{cases} U & \text{if } U \text{ is a prefix set} \\ (U\setminus UA^+)\cup(U^{-1}U\setminus\varepsilon) & \text{otherwise (indeed, } \varepsilon\in U^{-1}U).\end{cases}$$

By construction, $\sigma(U)$ is a set of suffixes of U, and we have $U\subseteq(\sigma(U))^*$. Moreover, since $\varepsilon\notin U$, we have also $\varepsilon\notin\sigma(U)$. Similarly, we shall define a corresponding function τ by truncating suffixes of U.

Let Z be a two-element subset of A^+ with $r(Z)>1$. By iterating function $\tau\circ\sigma$, we shall define a sequence $Z=Z_0,Z_1,\ldots$, such that $Z_i\subseteq(Z_{i+1})^*$. Since the sum of the lengths of the words in Z_i strictly decreases, there exists an integer k such that $Z_k=Z_{k+1}$. We set $T=Z_k$.

Since the sequence (Z_i) is stationary, T is a biprefix set, and if $r(Z)=2$ then we have $|T|=2$ (indeed we have $Z\subseteq T^*$).

Moreover, by construction, given a biprefix set $Y\subseteq A^+$, if $Z\subseteq Y^*$ then, for each integer $i\geqslant 0$, we have $Z_i\subseteq Y^*$; thus, we set $T\subseteq Y^*$. □

Let $X=\{w_1,\ldots,w_p\}\subseteq A^+$, with $p\geqslant 3$ and $r(X)\geqslant 2$. Let i be the smallest integer in $[2,p]$ such that w_1 and w_i have different primitive roots. Set $Z=\{w_1,w_i\}$. The construction in the proof of Lemma 3.2 leads to define an unique biprefix set T, and we set $ABSTRACT(X)=T$.

Clearly, $ABSTRACT(X)$ satisfies the following properties:

(3.1) (1) $ABSTRACT(X)$ is a biprefix set;

(2) $|ABSTRACT(X)|=2$;

(3) For all the biprefix sets Y, if $X\subseteq Y^*$ then $ABSTRACT(X)\subseteq Y^*$.

Example 3.3. Let $\{u,v\}$ be a biprefix set, and let

$$w_1=uvu,\qquad w_2=uvu^2vu,\qquad w_3=u^2vu,\qquad w_4=uvu^2.$$

Starting with $w=w_1$, we have $Z=\{w_1,w_3\}$ (indeed, $w_2=w_1^2$). Iterating $\tau\circ\sigma$ leads to compute the following sets:

$$Z_1=\{uvu,u\},\qquad Z_2=\{u,v\};$$

thus, we obtain $ABSTRACT(X)=\{u,v\}$.

3.2. Algorithmic interpretation

Given the set $X=\{w_1,\ldots,w_p\}\subseteq A^+$, we set $ABSTRACT(X)=T$. From an algorithmic point of view, the preceding construction leads to an algorithm for computing $ABSTRACT(X)$:

Algorithm 4: Function *ABSTRACT.*
begin
(1) apply the KMP-algorithm for computing the word $w_i\in X\setminus\{w\}$, with primitive root of $w_i\neq$ primitive root of w_1 and i minimal;
$Z\leftarrow\{w_1,w_i\}$;
(2) **while** $\tau\circ\sigma(Z)\neq Z$ **do** $Z\leftarrow\tau\circ\sigma(Z)$;
$ABSTRACT(X)\leftarrow Z$
end

Complexity of Algorithm 4

From an algorithmic point of view, words will be represented by linked lists. Recall that we set $n=\sum_{w\in X}|w|$, and $m=\max\{|w|:\ w\in X\}$.

Lemma 3.4. *Computing the set* $ABSTRACT(X)$ *will be done by applying Algorithm* 3 *in time* $O(n)$.

Proof.
- In step (1), computing the word w_i requires time $O(n)$.
- Applying functions σ and τ is done by comparing $L-L'$ letters, where L (L') stands for the sum of the lengths of the words in Z ($\sigma(Z)$). Consequently, given the set $Z=Z_0$, in step 2, computing $ABSTRACT(X)=Z_k$ will be done in time $O(\sum_{0\leqslant i\leqslant k-1} L_i - L_{i+1})$, with $L_i=\sum_{w\in Z_i}|w|$; thus, it requires time $O(n)$.
- As a consequence, applying Algorithm 4 requires time $O(n)$. □

4. Factorization of $ABSTRACT(X)$: the three main cases

Assume that $r(X)=2$. According to Proposition 2.1, throughout this section, we shall consider a set $Y=\{x,y\}$ which satisfies the following property:

(4.1) (1) Y is a biprefix set.

(2) x and y are primitive words.

(3) $X\subseteq Y^*$.

According to (3.1), we have also $ABSTRACT(X)\subseteq Y^*$. Clearly, one of the three following conditions holds:

(4.2) (1) $ABSTRACT(X)$ contains a word in $xyY^*\cap Y^*yx$ (or $yxY^*\cap Y^*xy$);

(2) $ABSTRACT(X)$ contains a word $w\in t^2Y^*\cup Y^*t^2$, with $t\in\{x,y\}$;

(3) $ABSTRACT(X)$ is included in $(xyY^*\cap Y^*xy)\cup(yxY^*\cap Y^*yx)$.

In each of these cases, we shall establish necessary conditions on the pair (x,y). Briefly, we shall construct different subsets of $A^*\times A^*$, namely P_i $(1\leqslant i\leqslant 2)$ and Q. These sets have cardinality $O(\ln^2 m)$; moreover, their elements leads to informations concerning (x,y). We start with the easiest case.

4.1. The case where a word in $ABSTRACT(X)$ *belongs to* x^2Y^*

In this section, we assume that there exists a word $w_1\in ABSTRACT(X)$ such that $w_1\in x^2Y^*$. In other words, the word x belongs to the set $SQUARE(w_1)$, whose elements are all the primitive words t such that $t^2\in pref(w_1)$. According to [4],

(4.3) Given a word $w\in A^*$, we have $|SQUARE(w)|\leqslant\log_\Phi|w|$, where Φ stands for the golden ratio $(1+\sqrt{5})/2$.

Moreover, given a pair of words $(w,t)\in A^*$, it is convenient to denote by $TRUNC(w,t)$ the shortest word w' which satisfies $w\in t^*w't^*$. With the preceding

condition, since $ABSTRACT(X)>1$, if $w_1 \in x^2 Y^*$, at least one of the two words $TRUNC(w_i, x)$ $(i=1,2)$ belongs to $yY^* \cap Y^* y$.

Now, we denote by P_{11} the set of all the pairs (t, w) which satisfy the condition:
- $t \in SQUARE(w_1) \cup SQUARE(w_2)$;
- w is the smallest nonempty word in $\{TRUNC(w_1, t), TRUNC(w_2, t)\}$.

Clearly, in the case where $ABSTRACT(X) \cap Y^* x^2 \neq \emptyset$, similar arguments on the reversed words allow the construction of a corresponding set P'_{11}. This leads to substitute to the actual set P_{11}, the set of all the pairs corresponding to the condition (4.2.2).

The following result comes from the preceding notation:

Lemma 4.1. *Given a finite set $X \subseteq A^*$, there exists a set $P_{11} \subseteq A^* \times A^*$ such that the following conditions hold:*

(1) $|P_{11}| \sim O(\ln m)$;

(2) *Given a set $Y=\{x, y\}$ which satisfies condition* (4.1), *with $ABSTRACT(X) \cap (x^2 Y^* \cap Y^* x^2) \neq \emptyset$, there exists a pair $(t, w) \in P_{11}$, such that $x=t$ and $w \in yY^* \cap Y^* y$.*

Example 3.3 (*continued*). Let $A=\{a, b, c\}$, $X=\{uvu, uvu^2vu, u^2vu, uvu^2\}$ with $u=(cbcaacbca)^3(cbca)^2$, $v=(aacbc)^2(acbc)^2$. We have $ABSTRACT(X)=\{u, v\}$, and

$$SQUARE(u)=\{cbcaacbca\}, \qquad SQUARE(\tilde{u})=\{acbc\},$$

$$SQUARE(v)=\{a, aacbc\}, \qquad SQUARE(\tilde{v})=\{cbca\}.$$

Moreover, we have

$$P_{11}=\{(cbcaacbca, (cbca)^2), (cbca, v)\}, (a, cbcaacbc(acbc)^2),$$

$$(aacbc, (acbc)^2), (acbc, aacbca)\}.$$

Algorithmic interpretation

After the computation of the function φ by applying Rule 1, the computation of *SQUARE* will be easily done by collecting all the prefixes u of the input word which satisfy $|\varphi(u)|=|u|/2$. It requires time linear in the length of the input word. Clearly, a similar result holds for the computation of *TRUNC*. As a consequence of (4.3),

(4.4) Given the set $ABSTRACT(X)$, computing P_{11} requires time $O(n \ln m)$.

4.2. The case where a word belongs to $xyY^* \cap Y^* yx$

Let $ABSTRACT(X)=\{w_1, w_2\}$. Without loss of generality, we assume that $w_1 \in xyY^* \cap Y^* yx$. Clearly, x is an overlap of w_1. First, we introduce a new notation.

Notation GENER( )

Given a nonempty word w, we denote by $GENER(w)$ the set of the pairs $(u, v) \in A^* \times A^+$, which satisfy the following conditions:
- uv is a primitive nonempty word

Fig. 2. A configuration of overlaps.

- given an overlap w' of w, there exists a unique pair $(u,v)\in GENER(w)$, such that $\varphi(w')v=u\varphi(w')$ (in other words, uv is the shortest word which satisfies $\{w',\varphi(w')\}\subseteq(uv)^+u$ (cf. Fig. 2).
  According to [6],

(4.5) Given a nonempty word w, we have $GENER(w)\leqslant\log_\Phi(|w|)$.

Since x is an overlap of w_1, there exists a pair of words $(u,v)\in GENER(w_1)$, such that $x\in(uv)^*u$. Let p be the greatest integer such that $(uv)^pu$ is both proper prefix and suffix of w_1. A theoretical study leads to more precise informations on the pair (x,y).

4.2.1. A theoretical study

Lemma 4.2. *Let $w\in x.pref(yY^*)\cap suff(Y^*y)x$. Let $u,v\in A^*$ such that uv is a primitive word, and such that $w=(uv)^pu$. If $x\in(uv)^iu$, with $1\leqslant i\leqslant p-2$, then only one of the two following conditions holds:*

(1) *there exists a unique integer k such that $y^k\in(vu)^*v$;*

(2) *$w\in x.pref(y)\cap suff(y)x$.*

Proof. Assume that $w\notin x.pref(y)\cap suff(y)x$. Two cases may occur.

Case 1: $w=xy^kx'$, with $x'\in pref(xY^*)$ and $k\geqslant 1$ (Fig. 3). One of the two following conditions holds.

(a) $x'\in x.pref(Y^*)$. Since uv is a primitive word, and a prefix of x, and according to (1.2), we have necessarily $y^k\in(vu)^*v$.

(b) $x'\in pref(x)$. If $|x'|\geqslant|uv|$ then $x'\in(uv)u$, and we conclude as in case a. Otherwise, x' is a proper suffix of vu, and according to the hypothesis $i\leqslant p-2$, we have $y^k\in(vu)^+pref(v)$. Similar arguments on the reversed words allow one to conclude that there exists an integer k' such that $y^{k'}\in suff(v)(uv)^+$. Since uv is a primitive word, we obtain $k=k'$ and $y^k\in(vu)^+v$.

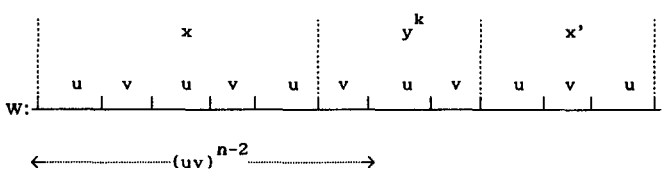

Fig. 3.

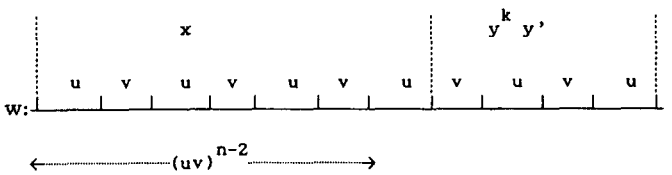

Fig. 4.

Case 2: $w=xy^ky'$, with $k\geqslant 1$ and $y'\in pref(y)$ (Fig. 4). Since y is a primitive word, two cases may occur.

(a) $y\in vu.pref(vu)^+$. Since $w\in suff(Y^*y)x$, and since $x\in(uv)^+u$, the word uv is a suffix of y; thus, we have $y\in(vu)^+v$.

(b) y is a prefix of vu. Since x is a prefix of $(uv)^{p-1}u$, the words $(vu)^2$ and y^{k+1} have a common prefix of length $2|vu|\geqslant|vu|+|y|$. Since Y is a biprefix, we have $y\neq vu$. According to Fine and Wilf's theorem, one of the two words vu and y is imprimitive, but this is a contradiction with the hypothesis of our lemma.

This completes the proof of Lemma 4.2. □

4.2.2. The consequences: necessary conditions on $\{x, y\}$

According to Lemma 4.2, with the condition of Section 4.2, only one of the three following conditions holds:

(4.6) (a) $x\in\{(uv)^{p-1}u,(uv)^pu\}$;

(b) $x\in(uv)^iu$ with $1\leqslant i\leqslant p-2$ and $y^k\in(vu)^*v$ $(k\geqslant 1)$;

(c) $x=(uv)^iu$ with $1\leqslant i\leqslant p-2$ and $y=(vu)^{p-i}.w'.(uv)^{p-i}$.

Consequences

We shall explicitly define a finite set $I(u,v)\subseteq[1,p]$, with $|I(u,v)|\leqslant 4$, and such that if one of the conditions (4.6) holds, then at least one of the following conditions also holds:

(4.7) (a) $x\in\{(uv)^iu\colon i\in I(u,v)\}$;

(b) $x\in(uv)^*u$ and $y^k\in(vu)^*v$ $(k\geqslant 1)$.

- Clearly, condition (4.6b) implies condition (4.7b).
- Assume that condition (4.6c) holds. If vu and uv are prefix and suffix of w', respectively, then the word $(uv)^{p+1}u$ is both prefix and suffix of w, but this is a contradiction with the maximality of the integer p. Hence, at least one of the two following conditions holds:
- vu is not a prefix of w'
- uv is not a suffix of w'

Assume that vu is not a prefix of w'. Since $ABSTRACT(X)$ is a prefix set, one of the two words $w'_1=(w_1\wedge w_2)^{-1}w_1$ and $w'_2=(w_1\wedge w_2)^{-1}w_2$, namely w, belongs to $(x\wedge y)^{-1}yY^*$. Since the words uv and vu are primitive and since they are prefix of

x and y, respectively, we have $x \wedge y = uv \wedge vu$. This allows to define the following integer i_1:

i_1 is the smallest of the integers $i \in [1, p-2]$ such that $(vu)^{p-i}$ is a prefix of one of the two words $(uv \wedge vu)w_1', (uv \wedge vu)w_2'$.

Similar arguments on the reversed words allow to define a corresponding integer i_2. Clearly, we have

$$x \in (uv)^i u, \text{ with } i \in \{i_1\} \cup \{i_2\}.$$

Let $I(u,v)$ be the set of those of the integers $p-1, p, i_1, i_2$ which are positive. As a consequence of the preceding properties, if one of the two conditions (4.6a), (4.6c) holds then we have:

$$x \in (uv)^i u, \text{ with } i \in I(u,v).$$

Conclusion

The preceding condition (4.7) leads to introduce the two following sets, namely P_{21}, Q_2:

(a) $Q_2 = GENER(w_1) \cup GENER(w_2)$;

(b) P_{21} is the set of the pairs of words $(t,w) \in A^+ \times A^+$, which satisfy the following conditions:

- t belongs to the union of the sets $\{(uv)^i u: i \in I(u,v) \cap \mathbb{N}\}$, for all $(u,v) \in Q_2$.
- w is a shortest nonempty word in $TRUNC(w_1, t) \cup TRUNC(w_2, t)$.

As a consequence, we have the following lemma.

Lemma 4.3. *Given the set X, there exists two sets $P_{21}, Q_2 \subseteq A^* \times A^*$ which satisfy the following properties:*

(1) $|P_{21}|, |Q_2| \sim O(\ln m)$;

(2) *Given a pair of words (x,y) such that $ABSTRACT(X)$ contains a word in $xyY^* \cap Y^*yx$, at least one of the following conditions holds:*

- *there exists a pair $(t,w) \in P_{21}$, with $x = t$ and $w \in yY^* \cap Y^*y$,*
- *there exists a pair $(u,v) \in Q_2$, with $x \in (uv)^*u$ and $y^+ \cap (vu)^*v \neq \emptyset$.*

Example 4.4. $ABSTRACT(X) = \{w_1, w_2\}$, with

$$w_1 = (cbcaacbca)^4 cbcaacbc \quad \text{and} \quad w_2 = (aacbc)^2 (acbc)^2$$

By iterating the function φ, which was defined in Section 1.2, we shall obtain the following overlaps of the word w_1:

- For each integer $i \in [1,3]$, $\varphi^i(w_1) = (u_i v_i)^{4-i} u_i$, with $u_i = cbcaacbc$, $v_i = a$;
- $\varphi^4(w_1) = u_4 v_4 u_4$, with $u_4 = cbc$, $v_4 = aa$;
- $\varphi^5(w_1) = u_5 v_5 u_5$, with $u_5 = c$, $v_5 = b$;
- $\varphi^6(w_1) = u_6 v_6 u_6$, with $u_6 = \varepsilon$, $v_6 = c$;
- $\varphi^7(w_1) = \varepsilon$.

Moreover, the only overlap of w_2 is ε.

According to the notation of Section 4.2, we have

$$Q_2 = \{(u_1, v_1), (u_4, v_4), (u_5, v_5), (u_6, v_6)\}.$$

Moreover, the elements of P_{21} are

$$\text{from } w_1\text{:}\quad ((u_1,v_1)^i u_1,(v_1u_1)^{4-i}) \quad (i=1,2,3),$$

$$\text{from } w_2\text{:}\quad (cbc,(aacbc)^2(acbc)a),\qquad (c,(aacbc)^2(acbc)acb).$$

4.2.3. Algorithmic interpretation

From an algorithmic point of view, computing $GENER(w)$ will be done by applying the following algorithm

Algorithm 5: Function *GENER*.
begin
$w'\leftarrow w$; $\lambda_0\leftarrow 0$; $GENER(w)\leftarrow\emptyset$;
(1) **while** $w'\neq\varepsilon$ **do**
begin
$w''\leftarrow\varphi(w')$; $\lambda\leftarrow|w'|-|w''|$;
(2) **if** $\lambda\neq\lambda_0$ **then**
begin
$v\leftarrow$the suffix of w' with $|v|=|w'|\bmod\lambda$ and $v\neq 0$;
$u\leftarrow$the prefix of w' with $|u|=\lambda-|v|$; $\qquad(*/\lambda=|uv|/*)$
$GENER(w)\leftarrow GENER(w)\cup\{(u,v)\}$; $\lambda_0\leftarrow\lambda$;
end;
$w'\leftarrow w''$
end
end

Complexity of Algorithm 5

- Loop (1) is applied $O(m)$ times. Moreover, in each operating cycle, the instructions outside of stage 2 require constant time.
- According to (4.7), the conditional stage 2 is applied $O(\ln m)$ times; thus, it requires time $O(m\ln m)$.
- Consequently, applying Algorithm 5 requires time $O(m\ln m)$.

By definition, we have $Q_2=GENER(w_1)\cup GENER(w_2)$. Moreover, the computation of P_{21} follows from the construction of Section 4.2.2.

Algorithm 6.
begin
$p_{21}\leftarrow\emptyset$;
for each integer $j\in\{1,2\}$ **do**
(1) **for** all the tuples $(u,v)\in GENER(w_1)$ **do** Q_2 **do**
begin
$p\leftarrow$the greatest integer with $(uv)^p u$ both prefix and suffix of w; compute $I(u,v)$ by directly applying its definition (Section 4.2.2);

```
      for all the integers i∈I(u, v) do
      begin
        t←(uv)^i u; w←TRUNC(w_j, t); P_21←P_21∪{(t, w)}
      end
    end
end
```

Complexity of Algorithm 6.

According to (4.7), loop (1) is applied $O(\ln m)$ times. Moreover, in each operating cycle of the loop, the corresponding operations require time $O(m)$. Consequently, computing the preceding sets P_{21}, Q_2 requires time $O(m \ln m)$.

4.3. The case where $ABSTRACT(X) \subseteq (xyY^* \cap Y^*xy) \cup (yxY^*yx)$

As in Section 4.2, we first establish a theoretical result.

4.3.1. Theoretical study

Lemma 4.5. *Let* $w \in xy\,pref(Y^*) \cap suff(Y^*)xy$. *Let* $u, v \in A^+$, *such that* uv *is a primitive word, with* $w=(uv)^p v$. *Assume that* $xy=(uv)^i u$, *with* $1 \leqslant i \leqslant p-2$. *If* uv *is a prefix of* x *then there exists a unique integer* k *such that* $y^k = v$.

Proof. If $x=uv$ then the word y belongs to $(uv)^*u$, a contradiction with the biprefixity of $\{x, y\}$. Hence, uv is a proper prefix of x. But since vu is a prefix of $(xy)^{-1}w$, we have $(xy)^{-1}w \in yY^*$.

Assume that vu is a prefix of y^k. Since y is a primitive word, and since it is a suffix of a word in $(vu)^+$, there exists an integer k' such that $y^{k'} = vu$. Since vu is also a suffix of xy, and since $\{x, y\}$ is a biprefix set, we obtain $k'=1$; thus, $y=vu$, and $x \in (uv)^*u$. But this is a contradiction with the biprefixity of $\{x, y\}$. Hence,

vu is not a prefix of a word in y^+.

Consequently, we have $w=(xy)y^k t$ (Fig. 5), with $t \in pref(xy^*)$. Moreover, since $i \in [1, p-2]$, and since uv is a primitive word, we have $y^k = v$.

This completes the proof of Lemma 4.5. □

4.3.2. The necessary conditions on (x, y)

As in Section 4.2.2, we now explain how the result of Lemma 4.5 leads to establish necessary conditions on the pair (x, y) which satisfies the conditions of Section 4.3.

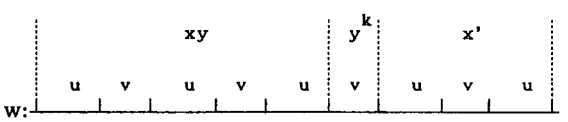

Fig. 5.

Let $ABSTRACT(X)=\{w_1, w_2\}$. Without loss of generality, we may assume that $w_1 \in xyY^* \cap Y^*xy$; there exists a pair of words (u, v), and a positive integer p, such that the following holds:

- $xy \in (uv)^+ u$, with uv a primitive word
- p is the greatest integer such that $(uv)^p u$ is both prefix and suffix of w_1.

As a consequence of Lemma 4.5, only one of the following conditions holds:

(4.9) (a) $xy=(uv)^i u$, with $i \in \{p-1, p\}$;
(b) $xy=(uv)^i u$, with $1 \leqslant i \leqslant p-2$, xy primitive, uv prefix of x, and $y^k=v$ $(k \geqslant 1)$;
(c) $xy=(uv)^i u$, with $1 \leqslant i \leqslant p-2$, xy primitive, and x proper prefix of uv;
(d) $xy=u^i$, with $2 \leqslant i \leqslant p-2$, and u a primitive word.

The consequences

We shall explicitly define two sets, namely P_{32} and Q_3, with $|P_{32}| \sim O(\ln m)$, $|Q_3| \sim O(\ln^2 m)$, and such that each of the conditions (4.9) implies one of the following:

(4.10) (a) there exists a word $w \in yY^* \cap Y^*y$, and an integer k, such that $((xy)^k, w) \in P_{32}$ $(k \geqslant 1)$;
(b) $X \subseteq \{xy, yx\}^*$;
(c) $x \in (u_1 v_1)^* u_1$ and $y^k \in (v_1 u_1)^* v_1$, or
$x^k \in (u_1 v_1)^* u_1$ and $y \in (v_1 u_1)^* v_1$, with $(u_1, v_1) \in Q$ and $k \geqslant 1$.

Indeed,

(a) With condition (4.9a), since Y is a code, the words xy and yx are different. According to the factorization of w_2 onto Y, two cases may occur:

(a.1) Assume that $w_2 \in xyY^* \cap Y^*xy$. Since $ABSTRACT(X)$ is a biprefix set, one of the words w_1, w_2, namely w, is not a power of xy. First, we introduce a new notation:

Given a word z and a set $T \subseteq A^*$, we denote by $LTRUNC(z, T)$ the shortest word such that $z \in T^+ . LTRUNC(z, T)$. In other words, $LTRUNC(z, T)$ is the shortest suffix of z obtained by left dividing elements of T. Let $w_0 = LTRUNC(w, xy)$. Two cases may occur.

(1) *The case where* $w_0 \in x^2 Y^*$. The word x belongs to $SQUARE(w_0)$; moreover, the word y belongs to $x^{-1}E$, with $E=\{(uv)^i u: i=p-1, p\}$. This leads to define the following set R:

R is the set of all the pairs $(r, r^{-1}t)$, which satisfies the two following conditions:
- $t \in E$;
- $r \in SQUARE(LTRUNC(w_1, t)) \cup SQUARE(LTRUNC(w_2, t))$ (clearly, since $r(ABSTRACT(X))=2$, we have $ABSTRACT(X) \setminus t^+ \neq \{\varepsilon\}$).

With this notation, if $w_0 \in x^2 Y^*$ then the pair (x, y) belongs to R.

Moreover, according to (4.3), we have $|R| \sim O(\ln m)$.

(2) *The case where* $w_0 \in yY^* \cap Y^*y$. We denote by S the set of all the pairs

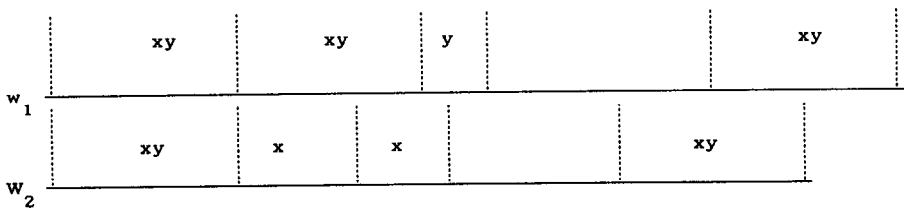

Fig. 6. Condition (4.9a) with $ABSTRACT(X) \subseteq xyY^*xy$; $w_1' = LTRUNC(w_1, xy) \in yY^* \cap Y^*y$, $w_2' = LTRUNC(w_2, xy) \in x^2Y$.

$(t, LTRUNC(w', t))$, which satisfies the two following conditions:

- $t \in E$;
- w' is the shortest nonempty word in $ABSTRACT(X) \setminus t^+$.

Clearly, we have $|S| \leqslant 2$. Moreover, the pair (xy, w_0) belongs to S (Fig. 6).

(a.2) Assume that $w_2 \in yxY^* \cap Y^*yx$; thus, the word yx is the prefix of w_2 with length $|xy|$. If $X \subseteq \{xy, yx\}^*$, then, clearly, we obtain the new two element "factorizing set" $\{xy, yx\}$.

If X is not included in $\{xy, yx\}^*$, let $w \in X \setminus \{xy, yx\}^*$, and let w' be the longest prefix of w in $\{xy, yx\}^*$. Clearly, the word $w_0 = TRUNC(w, \{xy, yx\})$ belongs to $x^2Y^* \cup y^2Y^*$. This leads one to define the following set R':

R' is the set of the pairs $(r, r^{-1}t)$, which satisfy the following conditions:

- $t \in E$ (E, being the set defined in case (a.1)
- $r \in SQUARE(LTRUNC(z, \{t, \bar{t}\}))$, where $\bar{t}$ stands for the prefix of w_2 with length $|t|$, and where z stands for the shortest nonempty word in $ABSTRACT(X) \setminus \{t, \bar{t}\}^+$.

According to (4.3), we have $|R'| \sim O(\ln m)$

(b) Now we assume that condition (4.9b) holds. Since y is a primitive word, it is the primitive root of v. We denote by u_1 the shortest word such that $u_1 y^{k+1} \in (uv)^+$, and we set $R'' = \{(u_1, v_1)\}$, with $v_1 = u_1^{-1}uv$.

The word xy^{k+1} belongs to the set $(uv)^+ = (u_1v_1)^+$. Moreover, since $u_1y^{k+1} \in (u_1v_1)^+$, we have $y^{k+1} \in (v_1u_1)^*v_1$, and $x \in (u_1v_1)^*u_1$ (cf. Fig. 7).

(c) By considering the reversed words, Conditions (4.9c) and (4.9b) are similar. With condition (4.9c), x is the primitive root of v, and we have $x^{k+1}y \in (vu)^+$. This leads to complete the preceding set R'' with a new pair of word (u_1, v_1), such that $x^k \in (v_1u_1)^*u_1$, and $y = (u_1v_1)^*u_1$.

(d) Assume that condition (4.9d), holds. By considering the reversed words, the case where $|y| \geqslant |x|$ and the case where $|y| \leqslant |x|$ may be examined in a similar way. Without loss of generality, we assume $|x| \geqslant |y|$. If u^2 is not a prefix of x then we have $xy = u^i$, with $i \leqslant 3$.

Assume that $i \geqslant 4$; thus, $u^2 \in pref(x)$. Clearly, there exists a pair of words, namely (u_1, v_1) such that $x \in (u_1v_1)^+u_1$, $y \in (v_1u_1)^*v_1$, and $u = u_1v_1$. We now give more precise informations about the word u_1.

(d.1) Assume that $w_2 \in xyY^*xy$ (Fig. 8). With the notation of case a, let $w \in ABSTRACT(X) \setminus u^+$ (indeed, $r(ABSTRACT(X)) = 2$). Clearly, one of the following conditions holds:

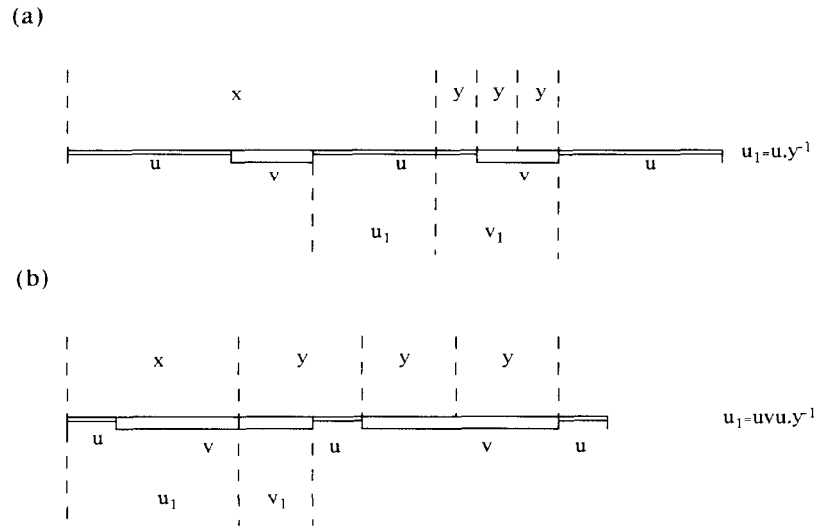

Fig. 7. Condition (4.9b) with $w \in X \setminus \{xy, yx\}^*$. (a) The case where $|y| < |u|$. (b) The case where $|y| \geq |u|$.

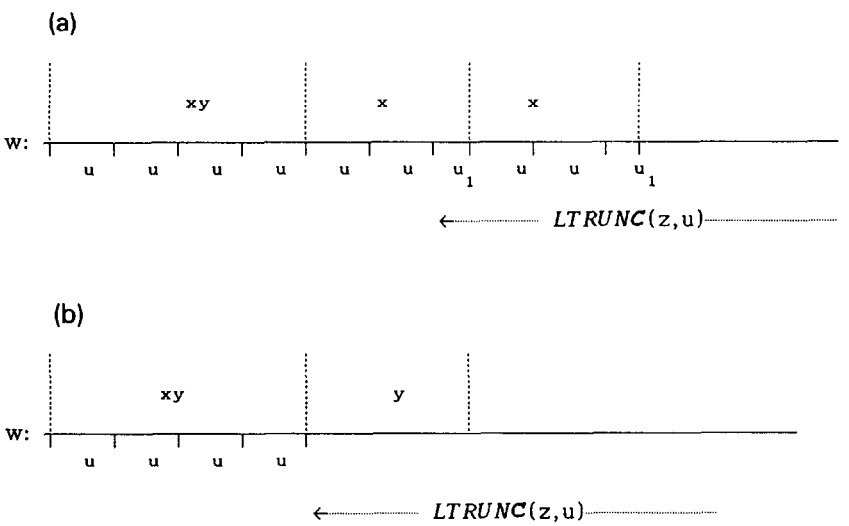

Fig. 8. $w_1, w_2 \in xyY^*xy$. (a) The case where $LTRUNC(w,u) \in x^2Y^*$. (b) The case where $LTRUNC(w,u) \in yY^* \cap Y^*y$.

(d.1.1) $LTRUNC(w,xy) \in x^2Y^*$; thus, the word w belongs to $u^+u_1u^2u^*A^*$. According to (1.1), since u is a primitive word, we have $u_1 = r$, where r stands for the shortest prefix of $LTRUNC(w,u)$ with ru^2 also a prefix of $LTRUNC(w,u)$. Let $R^{(3)} = \{(r, r^{-1}u)\}$.

(d.1.2) $LTRUNC(w,xy) \in yY^* \cap Y^*y$. As in the proof of Lemma 4.4, the word u cannot be a prefix of y. Hence, we have $LTRUNC(w,xy) = LTRUNC(w,u)$. Let $S' = \{(w, LTRUNC(w,u))\}$.

(d.2) Assume that $w \in yx\,Y^*\,yx$, and let $\bar{u}$ be the prefix of w_2 with length $|u|$. Clearly, u and $\bar{u}$ are conjugate words; moreover, we have $yx \in \bar{u}^+$, $u = u_1 v_1$ and $\bar{u}_1 = v_1 u_1$.

Since u is a primitive word, we have $u_1 = s$, with s the shortest prefix of u such that $s\bar{u}^2$ is a prefix of u^3. We set $R^{(4)} = \{(s, s^{-1}u)\}$.

Now we define the sets P_{32} (Q_3) as the union of the sets $S \cup S'$ $(R \cup R' \cup R'' \cup R^{(3)} \cup R^{(4)})$, for all the pairs $(u,v) \in GENER(w_1) \cup GENER(w_2)$.

According to the preceding study, if one of the conditions (4.9) holds then one at least of the condition (4.10) also holds.

Lemma 4.6. *Given the finite set X, there exists two sets $P_{32}, Q_3 \subseteq A^* \times A^*$, such that the following holds:*

(1) $|P_{32}| \sim O(\ln m)$, $|Q_3| \sim O(\ln^2 m)$;

(2) *Given a set $Y = \{x, y\}$ which satisfies condition* (4.1), *if* $ABSTRACT(X) \subseteq (xyY^* \cap Y^*xy) \cup (yxY^* \cap Y^*yx)$, *then one of the three following cases occurs:*

- $X \subseteq \{xy, yx\}^*$,
- *there exists a pair of words $(t,w) \in P_{32}$, with $xy \in t^+$ and $w \in yY^* \cap Y^*y$,*
- *there exists a pair of words $(u,v) \in Q_3$, with $x \in (uv)^*u$, and $y^k \in (vu)^*v$ $(k \geqslant 1)$.*

4.3.3. Algorithmic interpretation

From an algorithmic point of view, the results of Section 4.2.2 leads to algorithms for computing the sets P_{32}, Q_3. The computation makes beforehand use of the sets $GENER(w_i)$, which may be done in time $O(m)$ by applying Algorithm 5. After that, we shall successively apply the following algorithms [7A–7D].

Algorithm 7A.

begin

$P_{32} \leftarrow \emptyset$; $Q_3 \leftarrow \emptyset$; **for** each $j \in \{1,2\}$ **do**

(1) **for** all the tuple $(u,v) \in GENER(w_j)$ **do**

begin

$p \leftarrow$ the greatest integer such that $(uv)^p u$ is both prefix and suffix of w_j;

$I \leftarrow \{p-1, p\}$; **if** $v = \varepsilon$ **then** $I \leftarrow I \cup \{1,2,3\}$;

(2) **for** all $i \in I$ **do**

(3) **begin**

$t \leftarrow (uv)^i u$

if t is a prefix of w_2 **then** (*/ $w_2 \in xyY^*xy$ /*)

begin

for each word w in $ABSTRACT(X) \setminus t^+$ **do**

$P_{32} \leftarrow P_{32} \cup \{(t, LTRUNC(w, \{t\}))\}$

end

(4) **else begin** (*/ $w_2 \in yxY^*yx$ /*)

$\bar{t} \leftarrow$ prefix of w_2 with length $|t|$;

(5) **if** $X \subseteq \{t,\bar{t}\}^*$ **then** output('$r(X)=2$');
else begin
$w \leftarrow$shortest word in $ABSTRACT(X)\backslash\{t,\bar{t}\}^+$; $T=\{t,\bar{t}\}$
end
$w \leftarrow LTRUNC(w,T)$;
(6) **for** all $x \in SQUARE(w)$ **do** $Q_3 \leftarrow Q_3 \cup \{(x, x^{-1}t)\}$;
end
end
end
end

Complexity of Algorithm 7A

Clearly, applying function $LTRUNC$ with input (w, T) requires time $O(m)$ (indeed, since $|t|=|\bar{t}|$ it consists in at most $2|w|$ comparisons).

- According to (4.5), loop (1) is applied $O(\ln m)$ times.
- Clearly, loop (2) is applied at most 3 times (indeed, we have $|I| \leqslant 3$).
- Step 5 requires time $O(n)$,
- Since computing $x^{-1}t$ requires time $O(|t|)$, and according to (4.3), step 6 is done in time $O(m \ln m)$. As a consequence, stage 4 requires time $O(m \ln m)$.
- Consequently, applying Algorithm 7A requires time $O(m \ln^2 m)$

Algorithm 7B.
begin
(Q_3 has been first computed by applying Algorithm (7.A))
for each $j \in \{1,2\}$ **do**
(1) **for** all $(u,v) \in GENER(w_j)$ **do**
begin
(2) $y \leftarrow$primitive root of v; $k \leftarrow |v|$ div $|y|$;
$u_1 \leftarrow$shortest word such that $u_1 . y^{k+1} \in (uv)^*$; $v_1 \leftarrow u_1^{-1} uv$;
$Q_3 \leftarrow Q_3 \cup \{(u_1, v_1)\}$
end
end;

Complexity of Algorithm 7B

- After the computation of the values of function φ on the prefixes of v, step 2 is done in time $O(m)$ by applying the KMP-algorithm.
- Since loop (1) is applied $O(\ln m)$ times, Algorithm 7B runs in time $O(m \ln m)$.

Algorithm 7C. Similar to Algorithm 7B by considering the reversed words

Algorithm 7D.
begin
(Sets P_{32}, Q_3 has been first computed by applying Algorithms 7A–7C.)

(1) **for** all $u \in SQUARE(w_1)$ **do**
begin
if u prefix of w_2 **then** (*/ $w_2 \in xyY^*xy$ /*)
begin
$w \leftarrow$shortest word in $ABSTRACT(X) \setminus u^+$; $w \leftarrow LTRUNC(w, u)$;
$P_{32} \leftarrow P_{32} \cup \{(u, w)\}$
(2) $u_1 \leftarrow$shortest prefix of w with $u_1 u^2$ prefix of w; $v_1 \leftarrow u_1^{-1} u$;
end
else begin (*/ $w_2 \in yxY^*yX$ /*)
$\bar{u} \leftarrow$prefix of w_2 with $|\bar{u}| = |u|$;
(3) $i_1 \leftarrow$greatest integer with u^{i1} prefix of w_1;
$i_2 \leftarrow$greatest integer with u^{i2} prefix of w_2;
(4) $u_1 \leftarrow$shortest prefix of u with $u_1 . \bar{u}^2$ prefix of u^3; $v_1 \leftarrow u_1^{-1} u$;
end;
$Q_3 \leftarrow Q_3 \cup \{(u_1, v_1)\}$;
end
end

Complexity
- Loop (1) is applied $O(\ln m)$ times
- Step 2 is done in time $O(m)$ by applying the KMP-algorithm
- Step 3 consists only in comparing letters, and step 4 is done by applying the KMP-algorithm; thus, they require time $O(m)$.
- Consequently, applying Algorithm 7D will be done in time $O(m \ln m)$.

5. Conditions $\mathfrak{P}$ and $\mathfrak{Q}$

In the preceding section, we have collected informations on the sets $Y = \{x, y\}$, which satisfies condition (4.1). In view of obtaining a precise formalization of these informations, it is convenient to introduce two conditions, namely $\mathfrak{P}$ and $\mathfrak{Q}$.

(5) Let X be a finite set of words, and let P_i $(1 \leqslant i \leqslant 2)$ and Q, three finite subsets of $A^* \times A^*$. Let $Y = \{x, y\}$ be a biprefix set of primitive words.

Condition $\mathfrak{P}$. We say that Y satisfies condition $\mathfrak{P}$, with respect to the family $(P_i)_{1 \leqslant i \leqslant 2}$, iff there exists a word $w \in yY^* \cap Y^*y$ such that one of the two following properties holds:

($\mathfrak{P}1$) $(x, w) \in P_1$;

($\mathfrak{P}2$) there exists a word u such that $(u, w) \in P_2$ and $xy \in u^+$.

Condition $\mathfrak{Q}$. We say that Y satisfies Condition $\mathfrak{Q}$, with respect to the set Q, iff the following property holds:

There exists a pair $(u, v) \in Q_2$, and an integer $k \geqslant 1$, such that $x \in (uv)^*u$ and $y^k \in (vu)^*v$.

The results of Section 4, leads to the following lemma.

Lemma 5.1. *Given a finite set $X \subseteq A^*$, there exist three finite subsets of $A^* \times A^*$, namely $(P_i)_{1 \leq i \leq 2}$, and Q such that the following holds:*

(1) $|P_1|, |P_2| \sim O(\ln m)$, $|Q| \sim O(\ln^2 m)$;

(2) *For all the biprefix primitive sets $Y = \{x, y\} \subseteq A^*$, which satisfy Condition* (4.1), *one of the two following properties holds:*

- *Y satisfies condition $\mathfrak{P}$, with respect to $(P_i)_{1 \leq i \leq 2}$*
  Y satisfies condition $\mathfrak{Q}$, with respect to Q

(3) *Moreover, from an algorithmic point of view, the computation of (P_i) and Q may be done in time $O(n \ln^2 m)$.*

Proof. With the notations of Section 4, we set

$$P_1 = P_{11} \cup P_{21}, \qquad P_2 = P_{32}, \qquad Q = Q_2 \cup Q_3.$$

Then we get our lemma directly from the results of Section 4. □

6. A refinement of condition $\mathfrak{P}$ or condition $\mathfrak{Q}$

In this section, we shall establish a new necessary condition, which will be more concise than the preceding ones.

A first refinement lays upon the definition of a new set $Q' \subseteq A^* \times A^*$, such that the following holds:

- given a set Y which satisfies Condition $\mathfrak{P}$ or Condition $\mathfrak{Q}$, Y satisfies condition $\mathfrak{Q}$ with respect to Q'.

After that, an ultimate refinement will allow to define our set $TEST(X)$ (cf. Section 2.3).

6.1. A refinement of condition $\mathfrak{P}$

Let X be finite set of words, and let (P_i) be the family of finite subsets of $A^* \times A^*$ of Lemma 5.1.

Lemma 6.1. *There exists a set Q' such that the following holds:*

(1) $|Q'| \sim O(\ln^2 m)$;

(2) *Given a two-element set $Y \subseteq A^*$ satisfying condition* (4.1), *and Condition $\mathfrak{P}$ with respect to (P_i), then Y satisfies also Condition $\mathfrak{Q}$ with respect to Q'.*

Proof. Let $Y = \{x, y\}$ be a biprefix primitive set satisfying condition $\mathfrak{P}$. By definition, one of the two following cases occurs.

(i) *The case where condition* ($\mathfrak{P}1$) *holds*

With this condition, there exists a word $w \in yY^* \cap Y^*y$ such that $(x, w) \in P_1$. One of the three following cases occur:

(1) $w = y$; thus, $(x, y) \in T = \{(x, w)\}$.

(2) $w \in y^2 Y^*$; thus, $(x, y) \in T' = \{x\} \times SQUARE(w)$. Clearly, the case where $w \in Y^* y^2$ leads to define a corresponding set T'_1, and we substitute T' to $T' \cup T'_1$. According to (4.3), we have $|T'| \sim O(\ln m)$.

(3) $w \in yxY^* \cap Y^*xy$. As in Section 4.2, there exists a pair of words (u, v) such that $y \in (uv)^*u$. Let p be the greatest integer such that $(uv)^p u$ is both prefix and suffix of w. According to Lemma 4.2, one of the four following conditions holds:

(6.1) (a) $y \in \{(uv)^{p-1}u, (uv)^p u\}$.

(b) $y = (uv)^i u$ with $1 \leqslant i \leqslant p-2$ and $x^k \in (vu)^* v$ $(k \geqslant 1)$.

(c) $y = (uv)^i u$ with $1 \leqslant i \leqslant p-2$ and $x = (vu)^{p-i} . w' . (uv)^{p-i}$, with either vu not a prefix of w or uv not a suffix of w. Moreover, since $x \neq y$, we have necessarily $v \neq \varepsilon$.

We shall explicitly define a set $T'' \subseteq A^* \times A^*$, such that each of the preceding conditions (6.1) implies the following one:

(6.2) There exists a pair $(u_1, v_1) \in T''$ such that $y \in (u_1 v_1)^* u_1$ and $x^k \in (v_1 u_1)^* v_1$ $(k \geqslant 1)$.

(a) If condition (6.1a) holds, we have $x = u_1$, $y = v_1$, with $(u_1, v_1) \in T''_a = \{x\} \times \{u, (uv)^{p-1}u, (uv)^p u\}$.

(b) Clearly, with condition (6.1b), we directly obtain the terms of condition (6.2). Let $T''_b = \{(u, v)\}$.

(c) Assume that condition c holds.

Let i_1 (i_2) be the greatest integer j such that $(vu)^j$ ($(uv)^j$) is a prefix (suffix) of xy (yx). Clearly, the integer i in condition (6.1c) belongs to $\{i_1\} \cup \{i_2\}$. Let $T''_c = \{x\} \times \{(uv)^i u : i = i_1, i_2\}$;

Set $T'' = T''_a \cup T''_b \cup T''_c$. With this notation, and according to the preceding study, condition (6.1) implies condition (6.2). Moreover, we have $|T''| \leqslant 5$.

(ii) *The case where condition* ($\mathfrak{P}2$) *holds.* Let (t, w) be the corresponding pair of words in P_2. Recall that the word w belongs to $yY^* \cap Y^*y$, thus one of the two following cases occur:

(1) $w \in \{y\} \cup y^2 Y^* \cup Y^* y^2$. Given the word y, there exists a unique pair of words $(u, v) \in GENER(w)$ which satisfies the following condition:

(6.3) – the word u is the shortest word such that uy belongs to t^+
$uv = t$
– $x \in (uv)^* u$ and $y \in (vu)^* v$.

Let U be the set of the pairs (u, v) thus obtained for all the words $y \in \{w\} \cup SQUARE(w) \cup S$, where S stands for the set of the primitive words y with y^2 a suffix of w. We have $|U| \sim O(\ln m)$.

(2) $w \in yx\,Y^* \cap Y^* xy$. As in case (i.3), there exists a unique pair of words $(u, v) \in GENER(w)$ such that $y \in (uv)^* u$. We denote by p the greatest integer such that $(uv)^p u$ is both prefix and suffix of w. Clearly, condition (6.1) also holds. We shall define a new set U', such that a condition similar to (6.2) holds.

(a) As in case (ii.1), if condition (6.1a) holds, then we have $(x, y) \in U'_a$, where U'_a stands for the set of the pairs (u_1, v_1) which satisfy condition (6.3), for all the words y in $\{(uv)^i u \; i = p-1, \; p\}$. Clearly, we have $|U'_a| \leqslant 2$.

(b) With condition (6.1b) we directly obtain the terms of (6.2b). We set $U'_b = \{(u, v)\}$.

(c) Assume that condition (6.1c) holds; thus, we have

$$y = (uv)^i u \text{ with } 1 \leqslant i \leqslant p-2 \quad \text{and} \quad x = (vu)^{p-i} . w' . (uv)^{p-i},$$

with either vu is not a prefix of w' or uv is not a suffix of w'.

If vu is not a prefix of w' then, since $xy \in t^+$, we have $i = i_1$, with i_1 the smallest integer such that $(vu)^{p-i1}$ is a prefix of a word in t^+.

Assume that uv is not a suffix of w'. Since the words xy and yx are conjugate, we have $yx \in \bar{t}^+$, with t and $\bar{t}$ conjugate. Since $w \in yx\,Y^* \cap Y^* xy$, $\bar{t}$ is the prefix of w with length $|t|$. Moreover, we have $i = i_2$, with i_2 the smallest integer such that $(uv)^{p-i2}$ is a suffix of a word in $\bar{t}^+$.

Consequently, the word y belongs to the set $\{(vu)^i v\colon i \in \{i_1\} \cup \{i_2\}$. As in (ii.1), this leads to define the set U'_c, whose elements are the corresponding pairs of words $(u_{i1}, v_{i1}), (u_{i2}, v_{i2})$, which satisfies condition (6.3) for all the pairs $(u, v) \in GENER(w)$.

Set $U' = U'_a \cup U'_b \cup U'_c$. We have $|U'| \leqslant 5$. Moreover, with this notation, according to the preceding study, condition (6.1) implies condition (6.2).

Now we denote by Q'_1 (Q'_2) the union of the sets $T \cup T' \cup T''$ ($U \cup U'$), for all the pairs $(x, w) \in P_1$ ($(t, w) \in P_2$). Since we have $|T \cup T' \cup T''|, |U \cup U'| \sim O(\ln m)$, we obtain $|Q'_1|, |Q'_2| \sim O(\ln^2 m)$.

Moreover, we set $Q' = Q'_1 \cup Q'_2$. With this notation, if condition $\mathfrak{P}$ holds with respect to $(P_i)_{i=1,2}$ then Condition $\mathfrak{Q}$ also holds, with respect to Q'.

This completes the proof of Lemma 6.1. □

Algorithmic interpretation

From an algorithmic point of view, we shall construct the preceding set Q' by applying successively the following Algorithms 8 and 9:

Algorithm 8.
begin
$Q' \leftarrow \emptyset$;
(1) **for** all the pairs $(x, w) \in P_1$ **do**
begin
$Q' \leftarrow Q' \cup \{(x, w)\}$; $Q' \leftarrow Q' \cup \{x\} \times SQUARE(w)$;
(2) **for** all $(u, v) \in GENER(w)$ **do**
begin

$p \leftarrow$ the greatest integer with $(uv)^i u$ both prefix and suffix of w;
$Q' \leftarrow Q' \cup \{x\} \times \{(uv)^i u\colon i = p-1, p, i_1, i_2\}$; (*/cases 3a and 3c/*)
$Q' \leftarrow Q' \cup \{(u,v)\}$; (*/case 3b/*)
end
end
end

Complexity of Algorithm 8

- Since $|P_1| \sim O(\ln m)$, loop (1) is applied $O(\ln m)$ times.
- Applying function *GENER* requires time $O(m \ln m)$; moreover, according to (4.7), loop (2) is applied $O(\ln m)$ times, each operating cycle running in time $O(m)$.
- Consequently, applying Algorithm 7 requires time $O(m \ln^2 m)$.

Given a word $t \in A^*$, and a set E, we denote $PAIR(t, E)$ the set of the pairs (u, v) which satisfy condition (6.3) for all the words in E. From an algorithmic point of view, the corresponding function *PAIR* runs in time $O(|E| m)$.

Algorithm 9.
begin
(1) **for** all $(t, w) \in P_2$ **do**
begin
if $|w| > |t|$ **then** $\bar{t} \leftarrow$ the prefix of w with length $|t|$;
(2) $S \leftarrow \{\tilde{y}\colon y \in SQUARE(\tilde{w})\}$;
$S \leftarrow S \cup \{y\}$;
(3) $S \leftarrow SQUARE(w)$;
(4) **for** all $(u, v) \in GENER(w)$ **do**
begin
$p \leftarrow$ the greatest integer with $(uv)^i u$ both prefix and suffix of w;
$i_1 \leftarrow$ the smallest integer such that $(vu)^{p-i1} \in pref(t^+)$;
if $|w| > |t|$ **then**
$i_2 \leftarrow$ the smallest integer such that $(uv)^{p-i2} \in suff(\bar{t}^+)$
else $i_2 \leftarrow 0$;
$S' \leftarrow \{(uv)^i u\colon i = p-1, i_1, i_2\}$;
$S' \leftarrow S' \cup S$;
$Q' \leftarrow Q' \cup PAIR(t, S')$; $Q' \leftarrow Q' \cup \{(u, v)\}$
end
end
end

Complexity of Algorithm 9

- According to Lemma 5.1, loop (1) is applied $O(\ln m)$ times.
- Steps 2 and 3 requires time $O(m \ln m)$ (cf. Section 4).
- In each operating cycle (1), loop (4) is applied $O(\ln m)$ times. Moreover, the operating cycle (4) requires time $O(m)$.

– As a consequence, applying Algorithm 9 requires time $O(m \ln^2 m)$.

6.2. The ultimate condition and the proof of Theorem 2.2

Let us substitute Q by $Q \cup Q'$. According to Lemmas 5.1 and 6.1, we have $|Q| \sim O(\ln^2 m)$.

We are now able to complete the proof of Theorem 3.2. In fact, it consists in establishing a final refinement of condition $\mathfrak{Q}$.

Lemma 6.2. *Given a finite set $X \subseteq A^*$, and given the preceding set Q, there exists a set $TEST(X) \subseteq A^* \times A^*$ such that the following holds:*

(1) *For each pair $(\alpha, \beta) \in TEST(X)$, $\{\alpha, \beta\}$ is a biprefix set;*

(2) $|TEST(X)| \sim O(\ln^2 m)$;

(2) *Given a set Y satisfying condition* (4.1) *and condition $\mathfrak{Q}$ (with respect to the preceding set Q), there exists a pair $(\alpha, \beta) \in TEST(X)$, such that $Y \subseteq \{\alpha, \beta\}^*$.*

Recall that, with the first condition in Lemma 6.2, the flower automaton with behavior $\{\alpha, \beta\}^*$ is deterministic.

Proof. Let $Y = \{x, y\}$ be a two-element set which satisfies $\mathfrak{Q}$, with respect to Q. By definition, there exists a pair of words $(u, v) \in Q$, such that $x \in (uv)^* v$, and $y^k \in (vu)^* v$.

(1) If $k = 1$, then we have $Y \subseteq V^*$, with $V = ABSTRACT(\{u, v\})$ (cf. Section 3.2).

(2) Assume that $k > 1$. According to Fine and Wilf's theorem, we have $y^k \in \{v, vuv\}$.

– If $y^k = v$, then we have $Y \in \{u, r\}^*$, with r the primitive root of v. Let $V' = \{u, r\}$.

– Now, we assume that $y^k = vuv$. According to the lengths, if the primitive word y is a proper prefix of v, it is also a suffix of v. According to (1.2), we have $v \in y^+$; thus, $u \in y^+$, but this is a contradiction with Y being a code. Consequently, there exist two words, u_1 and u_2, with $u = u_1 y^{k-2} u_2$, and such that $y = u_2 v = v u_1$. Consequently (cf. e.g. [9, p. 8]), there exist two words α, β such that the following holds:

$$u_1 = \beta\alpha, \qquad v \in (\alpha\beta)^*\alpha, \qquad u_2 = \alpha\beta.$$

This implies that the word y belongs to $\{\alpha, \beta\}^*$. Moreover, since $u \in u_1 y^* u_2$, we have $u \in \{\alpha, \beta\}^*$; thus, $x \in \{\alpha, \beta\}^*$ (indeed $x \in (uv)^* u$). Once again, we have $Y \subseteq V''^*$, with $V'' = ABSTRACT(\{\alpha, \beta\})$.

(3) Now, we denote by $TEST(X)$ the union of the preceding sets V, V', V'', for all the pairs $(u, v) \in Q$. Clearly, the set $TEST(X)$ satisfies the conditions of Lemma 6.2. □

As a consequence, the set $TEST(X)$ satisfies the required conditions of Theorem 3.2. Moreover, the preceding construction leads to the following algorithm for computing $TEST(X)$.

Algorithm 10.
begin
$R \leftarrow \emptyset$;
for all $(u, v) \in Q$ **do**
begin
$R \leftarrow R \cup ABSTRACT(\{u, v\})$;
$y \leftarrow$ the primitive root of v; $R \leftarrow R \cup ABSTRACT(\{u, y\})$;
$y \leftarrow$ the primitive root of vuv; $u_2 \leftarrow yv^{-1}$;
$\alpha \leftarrow LTRUNC(v, u_2)$; $\beta \leftarrow \alpha^{-1} u_2$; $R \leftarrow R \cup ABSTRACT(\{\alpha, \beta\})$;
end;
$TEST(X) \leftarrow R$
end

Since $|Q| \sim O(\ln^2 m)$, and according to the results of Section 3 the computation of $TEST(X)$ requires time $O(n \ln^2 m)$.

References

[1] A. Apostolico and R. Giancarlo, Pattern matching implementation of a fast test for unique decipherability, *Inform. Process. Lett.* **18** (1984) 155–158.
[2] J. Berstel and D. Perrin, *Theory of Codes* (Academic Press, New York, 1985).
[3] J. Berstel, D. Perrin, J.F. Perrot and A. Restivo, Sur le théorème du défaut, *J. Algebra* **60**(1) (1979) 169–180.
[4] M. Crochemore and W. Rytter, Periodic prefix of strings, Act of Sequences '91, to appear.
[5] K. Culik II and J. Karhumäki, On the equality sets for homomorphisms on the free monoids with two generators, *RAIRO Inform. Théor. Appl.* **14** (1980) 349–369.
[6] J.P. Duval, Contribution à la combinatoire du monoïde libre, Thèse d'Etat, Université de Rouen, 1979.
[7] T. Harju and J. Karhumäki, On the defect theorem and simplifiability, *Semigroup Forum* **33** (1986) 199–217.
[8] D. Knuth, J. Morris and V. Pratt, Fast pattern matching in strings, *SIAM J. Comput.* **6** (1977) 323–350.
[9] M. Lothaire, *Combinatorics on Words*, Encyclopedia of Mathematics and its Applications (Addison-Wesley, Reading, MA, 1983).
[10] J. Néraud, Elementariness of a finite set of words is co-NP-complete, *RAIRO Theor. Inform. Appl.* **24** (5) (1990) 459–470.
[11] J. Néraud, On the deficit of a finite set of words, *Semigroup Forum* **41** (1990) 1–21.
[12] J. Néraud, On the rank of the subsets of a free monoid, *Theoret. Comput. Sci.* **99** (1992) 231–241
[13] D. Perrin, Finite automata, in: J. van Leeuwen, ed., *Handbook of Theoretical Computer Science, Vol. B* (Elsevier, Amsterdam, 1990) 1–57.
[14] M. Rodeh, A fast test for unique decipherability based on suffix trees, *IEEE Trans. Inform. Theory* **IT-28** (1982) 648–651.
[15] J.C. Spehner, Quelques problèmes d'extension, de conjugaison, et de présentation des sous-monoïdes du monoïde libre, Thèse de Doctorat d'Etat, Université de Paris VII, France, 1976.
[16] R. Capocelli and C. Hoffmann, Algorithms for factorizing and testing subsemigroups, in: A. Apostolico and Z. Galil, eds., Combinatorial Algorithms on Words (Springer, Berlin, 1984) 59–81.
[17] A. Lentin and M.P. Schützenberger, A combinatorial problem in the theory of free monoids, in: Combinatorial Mathematics and its Applications (Univ. of North Carolina Press, 1969) 128–144.